Amendments to the Claims:

Re-write the claims as set forth below. This listing of claims will replace all prior versions and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A graphics chip comprising:

a front-end in the graphics chip configured to receive one or more graphics instructions and to output a geometry;

a back-end in the graphics chip configured to receive said geometry and to process said geometry into one or more final pixels to be placed in a frame buffer;

wherein said back-end in the graphics chip comprises multiple parallel pipelines;

wherein said geometry is determined to locate in a portion of an output screen defined by a tile; and

wherein each of said parallel pipelines further comprises a unified shader that is programmable to perform both color shading and texture shading.

2. (canceled)

3. (canceled)

4. (previously presented) The graphics chip of claim 1 wherein each of said parallel pipelines further comprises:

a FIFO unit for load balancing said each of said pipelines.

5. (canceled)

6. (previously presented) The graphics chip of claim 1 wherein each of said parallel pipelines further comprises:

a z buffer logic unit; and

a color buffer logic unit.

7. (previously presented) The graphics chip of claim 6 wherein said z buffer logic unit interfaces with said scan converter through a hierarchical Z interface and an early Z interface.

8. (original) The graphics chip of claim 6 wherein said z buffer logic unit interfaces with said unified shader through a late Z interface.

9. (currently amended) A method for processing computer graphics comprising:

receiving one or more graphics instructions in a front-end of a graphics chip and outputting a geometry;

receiving said geometry in a back-end of the graphics chip;

processing said geometry into one or more final pixels to be placed in a frame buffer,

wherein said back-end comprises multiple parallel pipelines;

using a setup unit to direct said geometry into one of said multiple parallel pipelines;

wherein said geometry is determined to locate in a portion of an output screen defined by a tile; and wherein each of said parallel pipelines further comprises a unified shader and where the unified shader performs both color shading and texture shading based on programmable instructions.

10. (canceled)

11. (canceled)

12. (previously presented) The method of claim 9 further comprising:

using a FIFO unit for load balancing each of said pipelines.

13. (canceled)

14. (previously presented) The method of claim 9 wherein each of said parallel pipelines further comprises:

a z buffer logic unit; and

a color buffer logic unit.

15. (previously presented) The method of claim 14 wherein said z buffer logic unit interfaces with said scan converter through a hierarchical Z interface and an early Z interface.

16. (original) The method of claim 14 wherein said z buffer logic unit interfaces with said unified shader through a late Z interface.

17. (currently amended) A computer program product comprising:

a computer usable medium having computer readable program code embodied therein configured to process computer graphics, said computer program product comprising:

computer readable code configured to cause a computer to receive one or more graphics instructions in a front-end of a graphics chip and output a geometry;

computer readable code configured to cause a computer to receive said geometry in a back-end of a graphics chip;

computer readable code configured to cause a computer to process said geometry into one or more final pixels to be placed in a frame buffer,

wherein said back-end comprises multiple parallel pipelines;

computer readable code configured to use a setup unit to direct said geometry into one of said multiple parallel pipelines;

wherein said geometry is determined to locate in a portion of an output screen defined by a tile; and

wherein each of said parallel pipelines further comprises a unified shader that is programmable to perform both color shading and texture shading.

18. (canceled)

19. (canceled)

20. (previously presented) The computer program product of claim 17 wherein said computer readable code configured to cause a computer to process further comprises:

computer readable code configured to cause a computer to use a FIFO unit for load balancing each of said pipelines.

21. (canceled)

22. (previously presented) The computer program product of claim 17 wherein each of said parallel pipelines further comprises:

a z buffer logic unit; and

a color buffer logic unit.

23. (previously presented) The computer program product of claim 22 wherein said z buffer logic unit interfaces with said scan converter through a hierarchical Z interface and an early Z interface.

24. (original) The computer program product of claim 22 wherein said z buffer logic unit interfaces with said unified shader through a late Z interface.

25. (previously presented) The graphic chip of claim 1 comprising a setup unit for directing said geometry into one of said multiple parallel pipelines wherein said geometry is determined to locate in a portion of an output screen defined by a tile.

26. (previously presented) The graphics chip of claim 1 wherein each pipeline further comprises:

a scan converter;

a rasterizer; and

a texture unit.

27.(new) The graphics chip of claim 1 wherein the unified shader is operative to operative to apply a programmed sequence of instructions to rasterized values and is operative to loop back to process operations for color shading and/or texture address shading.

28. (new) The graphics chip of claim 1 wherein the unified shader is operative to apply a programmed sequence of instructions such that a first set of instructions are executed to perform color shading and a second set of instructions are executed to perform texture shading.

29. (new) The method of claim 9 wherein the unified shader applies a programmed sequence of instructions to rasterized values and loops back to process operations for color shading and/or texture address shading.

30. (new) The method of claim 9 wherein the unified shader applies a programmed sequence of instructions such that a first set of instructions are executed to perform color shading and a second set of instructions are executed to perform texture shading.